CLAIMS:

1. A method for managing variable sized pages of possibly non contiguous blocks in a Non-Volatile-Storage (NVS) for attaining a consistent NVS image that survives malfunction events; each page includes a self describing block or a linked list of self describing blocks, the method comprising:

(a) providing auxiliary modules stored in Volatile Storage;

(b) providing an atomic "create a new page" procedure;

(c) providing an atomic "add block" procedure for adding a possibly non contiguous block to a page; the newly added block has a back pointer to a previous block in the page;

(d) providing a "delete page" procedure for deleting all blocks in a page;

(e) providing at least one recovery procedure for rolling backward said add block procedure and rolling forward the delete page procedure, in case of malfunction event, thereby attaining consistent NVS.

2. The method according to Claim 1, wherein said auxiliary modules comprise free block database indicative of free blocks and association database representing replica of linked lists and partial linked lists, if any, of blocks in the NVS.

3. The method according to Claim 1, wherein said malfunction event being an electricity power malfunction.

4. The method according to Claim 1, wherein each block has the following data structure:

Block header that includes:

Block state : storing any of 'free'; 'used'; 'used-chained' values;

Entity identifier: storing entity identifier and applicable if state is not 'free';

Previous pointer: storing pointer to previous block in chain and applicable if state is 'used-chained';

Block user data: storing data that pertains to protected entity.

5. The method according to Claim 2, wherein said "add block" procedure comprises: applying atomic write that includes adding a block with a backward pointer to a previous block in the linked list of the page.

6. The method according to Claim 2, wherein said "delete page" procedure comprises:

while there are blocks in the page:

moving from a first block to last through forward pointers in the association database and deleting a corresponding block in the NVS; a block in the NVS is returned to the free list only after there is no block pointing to it.

7. A Non-Volatile-Storage (NVS) that includes variable sized pages of possibly non contiguous blocks; each page includes a self describing block or linked list of self describing blocks, using backward pointing scheme; said NVS is not susceptible to inconsistency in response to "create a new page", "add block to a page", or "delete blocks in a page" operations, irrespective of any intervening malfunction event.

8. The NVS according to Claim 7, wherein said malfunction event being an electricity power malfunction.

9. The NVS according to Claim 7, wherein each block has the following data structure:

Block header that includes:

Block state : storing any of 'free'; 'used'; 'used-chained' values;

Entity identifier: storing entity identifier and applicable if state is not 'free';

Previous pointer: storing pointer to previous block in chain and applicable if state is 'used-chained';

Block user data: storing data that pertains to protected entity.

10. The NVS according to Claim 7, having associated auxiliary modules stored in Volatile storage; the auxiliary modules comprise free block database indicative of free blocks and association database representing replica of linked lists and partial linked lists, if any, of blocks in the NVS.

11. The NVS according to Claim 10, wherein said "add block" procedure comprises: applying atomic write that includes: adding a block with a backward pointer to a previous block in the linked list of the page.

12. The NVS according to Claim 10, wherein said "delete page" procedure comprises:

while there are blocks in the page:

moving from first block to last through forward pointers in the association database and deleting a corresponding block in the NVS; a block in the NVS is returned to the free list only after there is no block pointing to it.

13. A system for managing variable sized pages of possibly non contiguous blocks in a Non-Volatile-Storage (NVS) for attaining a consistent NVS that survives malfunction events; each page includes a self describing block or linked list of self describing blocks, the system comprising:

Volatile Storage storing auxiliary modules;

means for performing an atomic "create a new page" procedure;

means for performing an atomic "add block" procedure for adding a possibly non contiguous block to a page; the newly added block has a back pointer to a previous block in the page;

means for performing a "delete page" procedure for deleting all blocks in a page;

means for performing at least one recovery procedure for rolling backward said add block procedure and rolling forward said delete page procedure, in case of malfunction event, thereby attaining consistent NVS.

14. The method according to Claim 1, for use in file systems that store meta-data on disk(s).

15. The Non-Volatile-Storage (NVS) according to Claim 7, for use in file systems that store meta-data on disk(s).

16. The system according to Claim 13, for use in file systems that store meta-data on disk(s).